AMENDMENTS TO THE CLAIMS

Please amend the Claims as follows:

Claims 1-32 (canceled).

33. (Currently Amended) An article of manufacture comprising having computer readable program code embodied therein which implements a method for compiling a structured document schema into type annotation records, said computer readable program code comprising:

a. computer readable program code building a type hierarchy ordered tree from structured document based on a derivation of relations among types in said structured document and determining one or more tuples for each type record in said structured document;

b. computer readable program code forming a complete typing set of said tuples;

c. computer readable program code sorting said typing set by their first field;

d. computer readable program code creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field and having a unique second field, collecting and sorting a third field from ambiguity typing sequences, assigning a unique offset number to each sorted third field, and arranging said ambiguity typing sequences based on offset numbers;

e. computer readable program code creating a typing array by concatenating typing tuples in resulting ambiguity typing sequences of (d);

f. computer readable program code for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple in N by typing tuple having offset, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence;

g. computer readable program code creating a type indexing data ~~structure and indicating ambiguity type for each type name; and~~, wherein entries in said type indexing data structure have a string field denoting element type, a flag field denoting ambiguity type for each type name, said flag field given a value of 'Y' is a corresponding element type is ambiguous and a value of 'N' when corresponding element type is not ambiguous, and an index field denoting a value corresponding to an index of an element type if corresponding flag field is set to 'N' or a value corresponding to a first index entry for an ambiguity sequence if a corresponding flag field is set to 'Y', said type indexing data structure being any of the following: hast table, binary tree, and B+ tree; and

h. computer readable program code writing said created index structure to storage.

34. (Original) The article of manufacture of claim 33, wherein said structured document schema is an XML document schema.

35. (Canceled).

36. (Canceled).

37. (Canceled).

38. (Canceled).

39. (Currently Amended) An article of manufacture comprising having computer readable program code embodied therein which implements a computer-based method for compiling a structured document schema into type annotation records, said computer readable program code comprising:

a. computer readable program code building a type hierarchy ordered tree from XML document schema based on a derivation of relations among types in said structured document and determining one or more tuples for each type record in said structured document

b. computer readable program code forming a complete typing set of said tuples;

c. computer readable program code alphabetical sorting said typing set by their first field;

d. computer readable program code creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field and having a unique second field, collecting and sorting a third field from ambiguity typing sequences, assigning a unique offset number to each sorted third field, and arranging said ambiguity typing sequences based on offset numbers;

e. computer readable program code creating a typing array by concatenating typing tuples in said resulting ambiguity typing sequences of (d);

f. computer readable program code, for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple in N by typing tuple having offset, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence;

g. computer readable program code creating any of the following type indexing data structures ~~and indicating ambiguity type for each type name: hash table, binary tree, or B+ tree; and~~, wherein entries in said type indexing data structure have a string field denoting element type, a flag field denoting ambiguity type for each type name, said flag field given a value of ‘Y’ is a corresponding element type is ambiguous and a value of ‘N’ when corresponding element type is not ambiguous, and an index field denoting a value corresponding to an index of an element type if corresponding flag field is set to 'N' or a value corresponding to a first index entry for an ambiguity sequence if a corresponding flag field is set to 'Y', said type indexing data structure being any of the following: hast table, binary tree, and B+ tree; and

h. computer readable program code outputting said created index structure.

40. (Canceled)

41. (Original) The computer-based method of claim 39, wherein said computer-based method is implemented in a database.